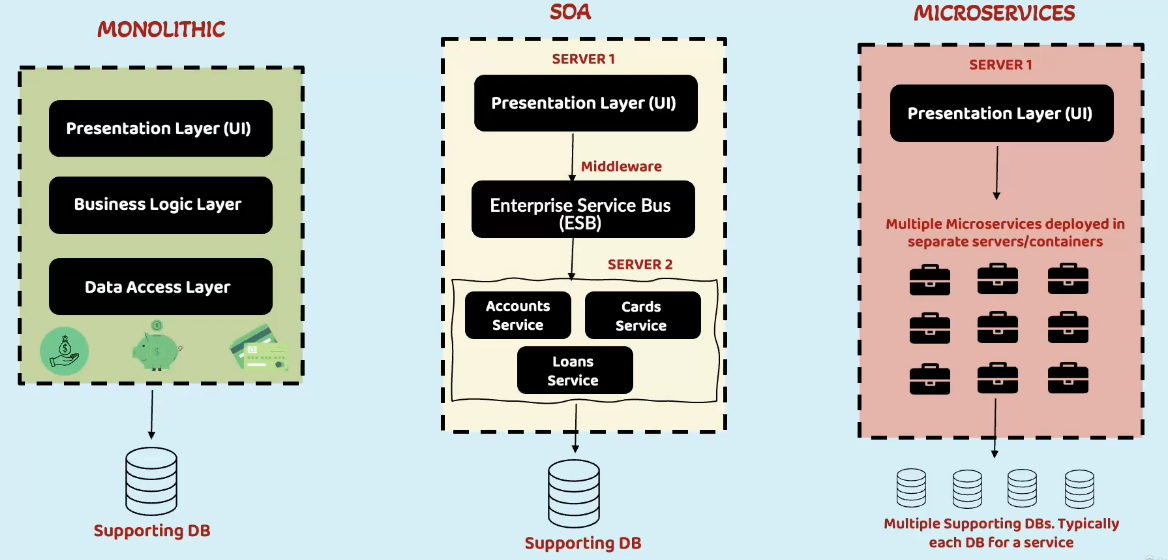
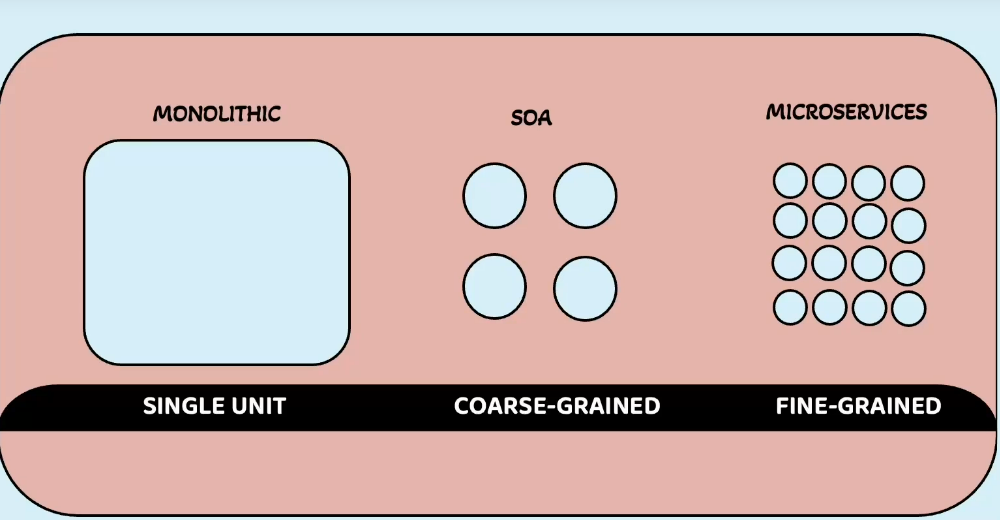
**Section 1  
Difference between Monolithic, SOA and Microservices - INTERVIEW**

Monolithic:   
All applications will be deployed in a single server, supported by a single database.  
The web application and the business logic is tightly coupled.  
  
SOA:  
We separate the UI and the backend logic in different servers, but again supported by a single database.   
We add a middle ware extra component (Enterprise Service Bus - ESB) which is difficult to maintain.  
There is some level of separation but not based on business domain.  
  
Microservice:  
Here we create separate deployable services based on business domain and each has its own database. All these microservices and databases can be a mix of technology.   
So, each service has its own separate development, code base and deployment lifecycle.  
  


|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **Monolithic** | **SOA** | **Microservices** |
| Parallel development by teams on features which are component specific | NO | MIDDLE (separation between UI and backend teams) | YES |
| Agility (Enhancing the application or component with new language / framework) | NO | MIDDLE (separation between UI and backend teams) | YES |
| Scalability | NO - Not possible because you are going to deploy all your application in one single jumbo server, so to scale you bring in one more jumbo server and need to take care of load balancing – manual setup | MIDDLE – Difficult to scale you, as are going to deploy all your application in one single jumbo server. | YES – Scalability is easy, because of products like docker and Kubernetes. |
| Usability | NO | MIDDLE | YES – Any new feature specific for domain component can be deployed without any interference to any other component. |
| Complexity and Operational Overhead | YES – Here there is only one server, you have to make sure just that one server is running without any issues | MIDDLE – Here you need to take care of UI server, middle ware server and backend server. | NO |
| Security and Performance | YES – Here there use to be method calls | MIDDLE | NO – Here the communication between components/service are REST services, so request goes over the network. So there may be some network latency. |

**Definition of Microservices**

**Section 2**